



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Internet of Things merupakan istilah yang umum digunakan untuk menggambarkan keterhubungan antar alat-alat yang sebelumnya tidak terkoneksi dengan Internet, kini menjadi terkoneksi dengan Internet dengan bantuan suatu perangkat kendali. Perangkat kendali tersebut tentunya akan memiliki perangkat lunak di dalamnya yang mengatur cara kerja alat tersebut dan hubungannya dengan alat-alat lain yang terkoneksi di dalam jaringan. Jaringan antar alat-alat ini umumnya berupa jaringan nirkabel yang terkoneksi dengan Internet. Seiring berkembangnya sistem, tentunya perangkat lunak yang ada pada perangkat kendali tersebut perlu diperbarui. Cara-cara untuk memperbarui perangkat lunak tersebut secara manual sangat tidak efisien dan menyulitkan *IoT Engineer* terutama ketika skala sistem sudah menjadi besar, sehingga cara-cara yang lebih baik sedang banyak dikembangkan. Cara tersebut adalah dengan perbaruan perangkat lunak melalui jaringan nirkabel, atau lebih dikenal dengan istilah *over-the-air update*.

Cara-cara untuk melakukan *over-the-air update* yang ada saat ini masih berupa solusi menggunakan layanan pihak ketiga yang berbayar, solusi yang *technology specific* [1], solusi yang kurang efisien untuk komunikasi nirkabel [2], serta solusi yang hanya memungkinkan *update* untuk 1 perangkat. Cara yang dimaksud adalah dengan menggunakan *library* ArduinoOTA pada *ESP Development Board Family* [3], atau dengan mengirimkan data ke *Raspberry Pi Development Board Family* melalui cara-cara umum seperti perintah *scp*. Solusi-solusi di atas merupakan solusi berbasis protokol TCP ataupun protokol yang terikat dengan teknologi tertentu. Hanya sedikit solusi yang menggunakan protokol UDP, padahal dari beberapa perbandingan, protokol UDP memiliki performa yang baik dari segi kecepatan dibandingkan dengan TCP ketika digunakan dalam pengiriman data, walaupun dibutuhkan tambahan *control* pada *application layer* untuk bisa

memberikan *reliability* yang menyerupai TCP [4] [5]. Selain itu, penggunaan protokol UDP juga memungkinkan distribusi data secara *multicast* untuk dilakukan.

Berangkat dari hal tersebut, penulis bertujuan untuk mengembangkan lebih lanjut penggunaan protokol UDP ini dalam *over-the-air update* dengan merancang arsitektur umum dari *over-the-air update system* yang berbasis UDP *Multicast* serta memiliki kemampuan untuk melakukan *one-to-many updates*. Diharapkan rancangan arsitektur umum ini dapat menjadi dasar yang baik dalam mengimplementasikan sistem *over-the-air update* berbasis UDP *Multicast* nantinya.

1.2. Identifikasi Masalah

Berdasarkan latar belakang masalah yang sudah dipaparkan, dapat diidentifikasi beberapa permasalahan yang dapat diteliti lebih lanjut yaitu:

- 1) Apakah *over-the-air update* berbasis UDP *Multicast* memiliki *success rate* yang sama dengan metode *update* yang serupa?
- 2) Apakah *over-the-air update* berbasis UDP *Multicast* dapat dilakukan lebih cepat dibandingkan dengan metode *update* yang serupa?
- 3) Apakah dapat diimplementasikan di beberapa *technology stack* yang umum digunakan dalam *IoT*?

1.3. Batasan Penelitian

Batasan masalah dari penelitian ini adalah sebagai berikut:

- 1) Sistem yang diimplementasikan hanya mencakup pendistribusian dari *updated code*, sedangkan implementasi proses *update* akan disesuaikan dengan masing-masing *end node*.
- 2) Sistem masih akan melakukan pengiriman *firmware* atau *code* sepenuhnya dan belum memiliki kemampuan untuk hanya mengirimkan perubahan dari kode.

- 3) Sistem yang diimplementasikan belum mempertimbangkan aspek *security* dari proses distribusi dan *firmware/code update* di *end node*.
- 4) Sistem yang diimplementasikan tidak menangani lebih lanjut bila terjadi *event device disconnect*.
- 5) Proses yang menangani *firmware* atau *code update* merupakan bagian dari *updated firmware* atau *code* dan harus memiliki akses terhadap *flash memory* serta *internal API* yang dibutuhkan untuk melakukan dan menyelesaikan *firmware* atau *code update*.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah mengimplementasikan sebuah sistem *over-the-air update* berbasis UDP *Multicast* yang dapat mendukung sistem *IoT* skala kecil hingga besar, dan membandingkannya dengan beberapa solusi *over-the-air update* dari aspek *success rate* dan *speed*.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

- 1) Memberikan acuan dasar bagi implementasi *over-the-air update* berbasis UDP *Multicast* yang mendukung *one-to-many updates*.
- 2) Mempermudah proses *mass deployment* untuk *IoT Systems*.

1.6. Sistematika Penelitian

Laporan penelitian ini disusun menjadi beberapa bagian untuk mempermudah pemahaman. Bab II menjelaskan mengenai penelitian-penelitian yang dijadikan referensi untuk penelitian ini beserta dengan poin-poin penting yang ingin digarisbawahi, serta deskripsi mengenai perangkat keras maupun perangkat lunak atau protokol yang digunakan dalam mengimplementasikan sistem untuk penelitian ini.

Bab III mendeskripsikan rancangan umum dari keseluruhan sistem, serta rancangan dari modul atau bagian dalam sistem. Selain itu, dijelaskan pula fitur dan spesifikasi singkat serta cara kerja dari sistem yang diimplementasikan dalam penelitian ini.

Bab IV berisi informasi mengenai implementasi dari sistem dari segi perangkat keras dan perangkat lunak berdasarkan rancangan pada Bab III. Kendala-kendala yang dihadapi dalam proses implementasi beserta solusi-solusi yang ditemukan juga akan tertera pada Bab IV. Setelah implementasi, skenario pengujian terhadap sistem serta hasil dari pengujian tersebut juga akan dicantumkan pada bagian ini.